Neural Networks 23 (2010) 905–916

Contents lists available at ScienceDirect

Neural Networks

journal homepage: www.elsevier.com/locate/neunet

# PAX: A mixed hardware/software simulation platform for spiking neural networks

S. Renaud [a,*], J. Tomas [a], N. Lewis [a], Y. Bornat [a], A. Daouzli [a], M. Rudolph [b], A. Destexhe [b], S. Saïghi [a]

[a] IMS, University of Bordeaux, ENSEIRB, CNRS UMR5218, 351 cours de la Libération, F-33405 Talence Cedex, France

[b] UNIC, CNRS UPR2191, 1 avenue de la Terrasse, F-91198 Gif-sur-Yvette, France

ARTICLE INFO

*Article history:*
Received 13 February 2008
Received in revised form 29 January 2010
Accepted 19 February 2010

*Keywords:*
Spiking neural networks
Integrated circuits
Hardware simulation
Conductance-based neuron models
Spike-Timing-Dependent Plasticity

ABSTRACT

Many hardware-based solutions now exist for the simulation of bio-like neural networks. Less conventional than software-based systems, these types of simulators generally combine digital and analog forms of computation. In this paper we present a mixed hardware–software platform, specifically designed for the simulation of spiking neural networks, using conductance-based models of neurons and synaptic connections with dynamic adaptation rules (Spike-Timing-Dependent Plasticity). The neurons and networks are configurable, and are computed in 'biological real time' by which we mean that the difference between simulated time and simulation time is guaranteed lower than 50 μs. After presenting the issues and context involved in the design and use of hardware-based spiking neural networks, we describe the analog neuromimetic integrated circuits which form the core of the platform. We then explain the organization and computation principles of the modules within the platform, and present experimental results which validate the system. Designed as a tool for computational neuroscience, the platform is exploited in collaborative research projects together with neurobiology and computer science partners.



## 1. Introduction

Computational neuroscience commonly relies on software-based processing tools. However, there are also various hardware-based solutions which can be used to emulate neural networks. Some of these are dedicated to the simulation of Spiking Neural Networks (SNN), and take into account the timing of input signals by precisely computing the neurons' asynchronous spikes. Neuron models can precisely describe the biophysics of spikes (action potentials) by computing the currents flowing through cell membrane and synaptic nodes. It is possible to reduce the size of these models to facilitate their computation. Other popular models are based on a phenomenological description of the neurons. They are well adapted to the study of complex network dynamics in neural coding or memory processing. While software tools can be configured for different types of models (Brette et al., 2007; Hines & Carneval, 1997) hardware-based SNN are dedicated to a given type of model. They may even be completely specialized, i.e. compute only one specific SNN model. In such a case, the hardware is designed for a specific application, as for example in the case of bio-medical artefacts (Akay, 2007).

Our group has been designing and exploiting neuromimetic silicon neurons for ten years (Renaud, Laflaquière, Bal & Le Masson, 1999, LeMasson, Renaud, Debay, & Bal, 2002, Renaud, Le Masson, Alvado, Saighi, & Tomas, 2004, Renaud, Tomas, Bornat, Daouzli, & Saïghi, 2007). We have developed specific integrated circuits (IC) from biophysical models following the Hodgkin–Huxley (HH) formalism, in order to address two fields of research: (i) build a hardware simulation system for computational neuroscience to investigate plasticity and learning phenomena in spiking neural networks; (ii) develop the hybrid technique, which connects silicon and biological neurons in real time. The system presented in this paper belongs to the first category, although it may be quite simply adapted for hybrid configurations. This platform was specifically designed for the simulation in biological real time of SNN using conductance-based models of neurons and synaptic connections: it enables the construction of bio-realistic networks, and offers the possibility of dynamically tuning the model parameters. The models are derived from the Hodgkin–Huxley formalism (Hodgkin & Huxley, 1952), and rely strongly on the physiological characteristics of neurons. The platform can run simulations of small networks of point neurons modeled with up to 5 conductances, using different cortical neuron model cards. Kinetic synapse models have been implemented to simulate the network connections. Each of the latter can be characterized by its own adaptation function, following a programmable rule for Spike-Timing-Dependent Plasticity (STDP). The SNN is computed in biological real time (the system ensures that the difference between simulation times and simulated time is less than 50 μs).

* Corresponding author.
*E-mail address:* sylvie.renaud@ims-bordeaux.fr (S. Renaud).

0893-6080/$ – see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.neunet.2010.02.006

In Section 2, we describe the various models classically implemented on hardware simulators, and review various simulation platforms. We discuss the relationship between the SNN formats and the architecture of the platforms, and discuss our technical solution. In Section 3 we provide a detailed description of the architecture of the simulation platform (PAX), and present its specifications and key features. In Section 4 we present experimental results obtained using PAX: benchmark simulations to validate the implemented hardware, and open experiments to study biologically relevant SNN. Finally, we provide our conclusions and present the future versions of the PAX platform: technical evolutions of the various elements, and experiments to be implemented in a collaborative project between physicists and biologists.

## 2. Hardware-based SNN platforms

There are various ways in which SNN models can be computed, ranging from software to hardware implementations. Dedicated software tools are well known (Brette et al., 2007) and widely distributed. Although offering numerous models and parameters, they often have the drawback of requiring prohibitively long computation times when it comes to simulating large and complex neural networks. Recent improvements have been achieved using parallel and distributed computation (Johansson & Lansner, 2007; Migliore, Cannia, Lytton, Markram, & Hines, 2006). These systems use a large computation power to process complex networks rather than guarantee real-time performance. Another approach is to build dedicated hardware to process a network of predefined neuron types. Hardware approaches, like the one we describe in the present paper, have very interesting properties in terms of computation time but a higher development cost and constraints on the computed models. Therefore these systems are generally application-dedicated. Applications can range from purely artificial experiments, in particular the investigation of adaptation and plasticity phenomena in networks, to experiments on hybrid biological/artificial networks.

For such systems, designers generally consider simplified neuron models and balance imprecision by a higher number of neurons in the network (Fieres, Schemmel, & Meier, 2006; Indiveri, Chicca, & Douglas, 2006). An other approach is to reduce the number of the parameters in the system (Farquhar & Hasler, 2005), or to model neurons populations (Fieres et al., 2006; Indiveri et al., 2006; Renaud et al., 2004). Considering the network structure, some systems limit synaptic connectivity, others guarantee 'all-to-all' connected networks with uniform connection delays, but with less neurons. Finally, systems differ by their properties in terms of computation time, as some systems aim to simulate SNN "as fast as possible" (Indiveri et al., 2006), and other guarantee a fixed simulation timescale (Bornat, Tomas, Saïghi, & Renaud, 2005; Fieres et al., 2006).

In the following section we provide a summary of the principal models used for spiking neurons and synaptic connections. We describe several types of hardware-based SNNs and compare it to ours; this review highlights the diversity of possible solutions, and their respective relevance to various applications.

### 2.1. The model choice issue

#### 2.1.1. The neuron level

Most SNN models are point neuron models, in which a neuron represents a single computational element, as opposed to compartmental models which take into account the cells' morphology. Different levels of abstraction are possible, to describe point neuron models. One can focus on the properties of each ionic channel, or prefer to choose a behavioral description. The user needs to select the model by finding an acceptable compromise between two contradictory criteria: faithfully reproduce the membrane voltage dynamics ($V_{MEM}$), and minimize the computational load on the simulation system.

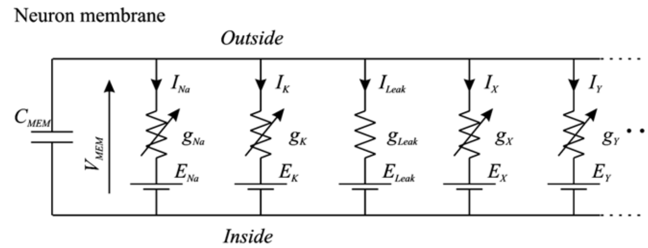


**Fig. 1.** Equivalent electrical circuit for HH-based neuron models. In this example, additional conductances ($x, y, \ldots$) modulate the influence of the Na, K and leak channels.

In the most detailed family of models, known as conductance-based models, ionic and synaptic currents charge and discharge a capacitor representing the neuron membrane (Gerstner & Kistler, 2002). All of these models find their origins in the Hodgkin and Huxley (1952) model (HH). Each ionic channel (Sodium: Na, Potassium: K…) is represented by a time- and voltage-dependent conductance: this electrophysiological description makes these models particularly well suited to an implementation involving analog electronics. Hodgkin–Huxley derived models have the same structure and include a larger number of types of ionic channel, in order to fit the fast and slow dynamics of the neurons. This multiplicity of models enables the diversity of biological neurons to be suitably represented (see Fig. 1). The main advantage of this formalism is that it relies on biophysically realistic parameters and describes individual ionic and synaptic conductances for each neuron in accordance with the dynamics of ionic channels. This type of model is necessary to emulate the dynamics of individual neurons within a network.

Conductance-based models reduced to 2 dimensions are also very popular, as they can be entirely characterized using phase plane analysis. The well-known FitzHugh–Nagumo (FN) (FitzHugh, 1961) and Morris–Lecar models (Morris & Lecar, 1981) are also worthy of mention.

Threshold-type models are another class of widely used models for SNN. They describe, at a phenomenological level, the threshold effect in the initiation of an action potential. The shape of the $V_{MEM}$ signal is not reproduced by such models, and ionic currents are no longer processed. These models are adjusted by fitting the timing of the spikes and setting the threshold level. In terms of computational cost, they are interesting for the study of neural coding and the dynamics of large networks. The Integrate-and-Fire model (IF) and Spike Response Model (SRM) belong to this family (Gerstner & Kistler, 2002). The Leaky Integrate-and-Fire model (LIF), which although simple is able to compute the timing of spikes, is widely used for hardware SNN implementations (see Fig. 2). The LIF model computes $V_{MEM}$ on a capacitor, in parallel with a leak resistor and an input current. When $V_{MEM}$ reaches a threshold voltage, the neuron fires and its dynamics are neutralized during an absolute refractory period. Many models were derived from the LIF, and take into account (for example) the effects of modulation currents (Brette & Gerstner, 2005; Gerstner & Kistler, 2002; Izhikevich, 2004).

#### 2.1.2. The network level

The dynamics of a SNN and the formation of its connectivity are governed by synaptic plasticity. Plasticity rules formulate the modifications which occur in the synaptic transmission efficacy, driven by correlations in the firing activity of pre- and post-synaptic neurons. At the network level, spikes are generally